UNIVERSITY OF QUEENSLAND
COMPUTER CENTRE

COMPUTER CENTRE BULLETIN

Volume 3, Number 10.
5th October, 1970.

Editor:
S. . Barry.

WEEKLY NEWSLETTER

It has long been felt that the Bulletin, because it is published monthly, often fails in bringing programming information to the early attention of computer users. The Computer Centre has therefore decided to produce a weekly newsletter for the benefit of its clients. This report will contain details of operations as well as various items of information pertinent to programming and computer services. It will be made available in the foyer of the Centre each Friday afternoon, (if Friday is a public holiday the Newsletter will be published on the last working day of the week). It will not be distributed by mail. All programming information published in this weekly newsletter will be consolidated and included in the Bulletin each month.

OPERATIONS SCHEDULE BOARD

A notice board detailing a Schedule of Operations for the PDP-10 is to be erected in the clients' room as soon as possible. The purpose of this is to outline the availability of the system for the current week. This will provide users with a guide for planning their runs and should result in more efficient use of the system.

EXTENDED DATA PREPARATION SERVICES

The data preparation service presently exists on a 'first come first served' basis. As from Monday 5 October 1970 the service is being extended to include advance bookings of keypunching time. This is in addition to the existing system and will operate in parallel with it.

The conditions under which block bookings of keypunching time will be accepted are as follows:

(i) The minimum block of time which can be booked under these arrangements is 15 minutes.

(ii) Bookings should be made with the Administrative Officer at least 24 hours in advance of the desired time, but availability of time cannot be guaranteed.

(iii) Charges will be at the standard rates for the length of time booked.

(iv) Advice of cancellation of a booking must be received at least 24 hours prior to the allocated booking. Cancellations made later than this will result in a charge being made for the period booked.

(v) Material for keypunching must be submitted on standard Computer Centre coding forms accompanied by the appropriate Requisition for Data Preparation form.

(vi) Keypunching will be terminated at the conclusion of the period booked whether the work is completed or not. The user may then make arrangements for keypunching the balance of his work.

All queries concerning data preparation services should be directed to the Administrative Officer, Mr John Jauncey, on extension 8471.

CHRISTMAS SHUT DOWN

The Computer Centre will shut down for the Christmas period at midnight on Wednesday 23 December 1970. There will be no processing of work on the 24 December to allow the staff to finalise all end of year operations and accounting. The Centre will as usual be closed over the Christmas - New Year period and will re-open for business at 9 a.m. on Monday 4 January 1971.

PDP-6 PROGRAMS

Mr N. A. Campbell of the CSIRO Mathematical Statistics Department at Wembley W. A. has generously offered to make available to University of Queensland Computer users the programs described below. These programs run on the CSIRO's PDP-6 computer but should be readily adaptable to the PDP-10 at the Computer Centre. Unfortunately, because of present commitments on remote terminal work, the Centre is unable, at present, to offer any support in respect of these programs other than via the normal consulting service. It must be the responsibility of the user to make any modifications which may be required to adapt these routines to the PDP-10. Anyone interested in using the programs should contact Dr D. J. Connor at the Botany Department, ext. 773.

1. PROGRAM NAME: PRLCPT Principal Components Analysis

INPUT: Card Reader

OUTPUT: Line Printer

DESCRIPTION:

PRLCPT computes the eigen values and vectors of a square symmetric matrix by the Householder method.

The original data can be read in, or the variance-covariance matrix can be read in. Data may be transformed if required.

2. PROGRAM NAME: CANCOR Canonical Correlation

INPUT: Card Reader

OUTPUT: Line Printer

DESCRIPTION:

CANCOR provides a canonical correlation analysis as described in Anderson: An Introduction to Multivariate Statistical Analysis p. 288 and A.T. James: Multivariate Statistical Analysis of Atmospheric Processes pp. 26-51. The program allows for a maximum of 20 predictor (independent) variables and 20 predictand (dependent) variables. Transformation of the data is available.

3. PROGRAM NAME: MRGD Multiple Regression

INPUT: Card Reader

OUTPUT: Line Printer

DESCRIPTION:

MRGD is designed for multiple linear, simple linear, and curvilinear regression. Variables may be transformed by the use of transformation cards. Up to 15 variables (dependent, independent and transformed) may be stored in the program.

MRGD eliminates variables in the reverse order to which they are fitted (i.e. last variable is eliminated if not significant, etc.) until a variable with a significant variance ratio is reached (i.e. when the mean square due to the particular variable is significantly greater than the residual mean square).

4. PROGRAM NAME: MULDIS Multiple Discrimination

INPUT: Card Reader

OUTPUT: Line Printer

DESCRIPTION:

MULDIS provides a multiple discrimination analysis as described in Seal: Multivariate Statistical Methods for Biologists pp. 123-152 - allows for up to 20 groups with up to 20 variables (measurements) per group. Transformation of the data is available.

PDP-10 BATCH CHANGES

On Monday, 14 September, a new version of the Batch processing control program was released for general use. The main purpose of the release was to improve efficiency and overcome errors previously reported. While every attempt has been made to ensure that errors have been eliminated, it is possible that some errors have not been corrected or further errors may have been introduced. The complexity of the major elements of the system is such that it is not possible to test for all possible error conditions. We would ask your help in reporting suspected errors promptly, so that they may be investigated and corrected as soon as possible.

It is the user's responsibility to include some degree of consistency checking into his programs to guard against the possibility of incorrect results caused by hardware or software errors. The user should also be aware that discrepancies may occur, and particularly after a system change should check runs carefully for discrepancies.

FORTRAN COMPILER ERROR

In attempting to optimize the code produced when an expression involves common sub-expressions, the following case is not handled properly:

```
A = (1.0 - X - Y)/(1.0 - X)
```

The code produced actually evaluates:

```
            A = (1.0 - X - Y)/(1.0 - X - Y)
           [  = 1.0                        ]
```

To program around this, separate the evaluation of the numerator and denominator or define the function in a way that does not involve the common sub-expression. In the example quoted:

```
            A = 1.0 - Y/(1.0 - X)
```

would be satisfactory.

This problem has been cured in DEC's version 20 of the FORTRAN compiler, which will be released for general use when it has been tested.

GE-225 FORTRAN IV ERROR

In a subroutine, when handling references to local arrays, the compiler attempts to create code that puts the variable in the subscript expression in an index register and inserts the constant portion of the subscript expression into the LDA instruction which accesses the data. Under some circumstances, it is possible that the calculated offset is negative, and in this case *subsequent* array references are compiled incorrectly.

In the following example:

```
            SUBROUTINE EXAMPL (ICH)
            DIMENSION ICH(80),ICI(8)
            .
            .
            .
            I=64
       10   ICH(I)=ICI(I-62)
            .
            .
            .
```

Statement 10, while actually legitimate, is likely to create a negative offset in the LDA instruction that accesses the element of ICI and all subsequent array references are liable to be incorrect.

The problem can be cured by evaluating the offending subscript expression prior to its use in the array reference.

SEMINARS IN COMPUTER SCIENCE

The last seminar scheduled for the year will be held on Wednesday, 14 October in Room 214, Engineering Administration Building.

The seminar will commence at 2.00 p.m. and finish no later than 4.00 p.m. The speaker will be Mr R. E. Kelly, Senior Lecturer in the Department of Computer Science, who will discuss *File Organization in Timeshared Systems*. The following abstract outlines the theme of the seminar.

An essential requirement of any timesharing computer is a general system for the storage of users' files. These files may be programs in source, relocatable, or absolute form; or they may be data files which are organised in a variety of ways. Not only must these files be readily accessible but the storage capacity of the file system must be effectively infinite. Factors which influence the accessibility of files include the number and characteristics of on-line users; the type, speed and capacity of the storage devices comprising the file system; and the software which controls access to it. The first two factors are largely invariable but the third has a profound effect on the efficiency, flexibility and reliability of the system.

The lecture will review some commonly used systems of file storage and will discuss some of the basic techniques of file organization. It will be argued that the system should be regarded as a single-level direct-access store with automatic file migration between storage devices. Further, the manner in which files are structured is of major importance.

FEATURES OF ATLAS - HARDWARE AND SUPERVISOR

M. J. McLean

Mr McLean is a Senior Demonstrator in the Department of Computer Science. He joined the Department in January this year. After graduating from The City University, London, with a B. Sc. in Civil Engineering, Mr McLean obtained his M. Sc. in Computer Science at the University of London's Institute of Computer Science. It was at the Institute that Mr McLean gained his experience with the Atlas computer.

This article has been adapted from a seminar given during the second term of this year. Owing to the length of the article it will be divided into two sections dealing with the hardware and software. Next month's Bulletin will carry the description of the software features of Atlas.

The Atlas system (designed and built in the late 1950's) was a pioneer in the field of fully automatic operating systems. Although it is purely a batch processing system the lessons learned have been incorporated in many modern timesharing systems. In particular, the techniques of Paging and One-Level-Store have been incorporated in the IBM 360/67 and the GE-645.

Each machine instruction occupies one 48-bit word. The format is quite conventional in that it specifies an operation-code, two General Purpose Registers and an address part (Fig. 1). Each 7-bit General Purpose Register field specifies one of the 128 General Purpose Registers. The 21-bit address part permits reference to two million unique addresses called the VIRTUAL MEMORY. Of this the user is only allowed to use the first million words and the other million is reserved for the Supervisor.

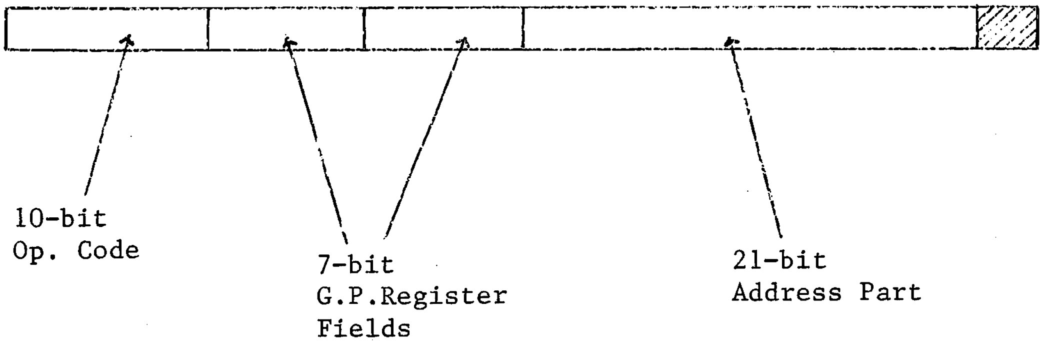

Figure 1 The Format of an Atlas Instruction

Every user program can be written as though it is to be run in an otherwise empty machine with a million word core store. This illusion is maintained by two parts of the system: the Paging hardware and the One-Level-Store software. The Paging hardware causes the 21-bit Virtual Memory addresses to be mapped successfully onto the 32K Actual Memory. The One-Level-Store software keeps most of the Virtual Memory on fast-drum backing store. In practice the total amount of Virtual Memory which can be used is a little less than 128K. This quantity is limited by the size of the backing store. Sections of Virtual Memory are automatically brought into core when they are needed.

The Virtual Memory is logically divided into 4096 blocks, each containing 512 words. The block is the basic unit of Virtual Memory and is the unit manipulated by the One-Level-Store routines. The 21-bit address can be considered as consisting of a 12-bit block number and a 9-bit word number which specifies the position of the word within the block (Fig. 2).

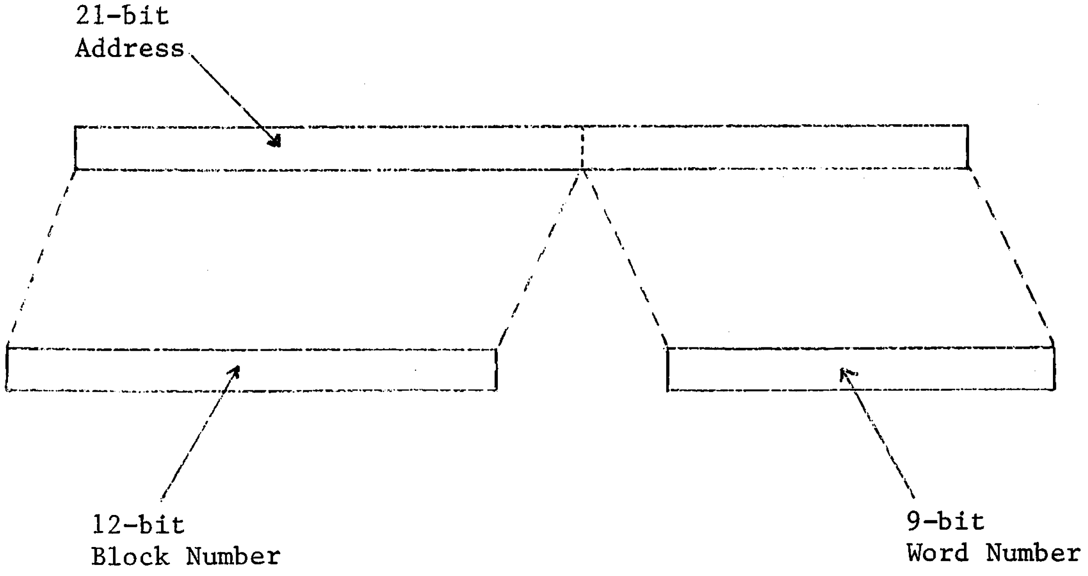

Figure 2 Subdivision of a 21-bit Logical Address

The 32K core memory is divided into 64 pages, each containing 512 words. It is worth noting the distinction between blocks and pages. A block is a logical quantity. It is 512 words of Virtual Memory and consists of program or data or both. A page is a physical entity. It is 512 words of contiguous words of core memory. Blocks are numbered 0 - 2047, pages 0 - 63. The 15-bit address necessary to address the 32K core can also be considered as consisting of two fields. Six bits specify the page number and the remaining nine bits specify which of the 512 words in that page is to be used (Fig. 3). Whenever a block is in core it occupies one page. If block B occupies page P, then word W of block B occupies word W of page P. This means that to locate a logical address in physical core, it is merely necessary to replace the block number in the address with the corresponding page number. This substitution is automatically performed by the Paging Unit.

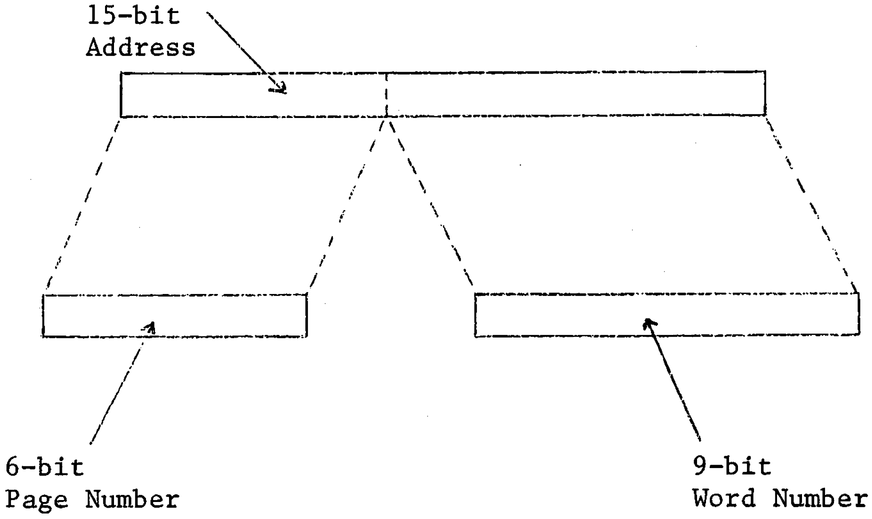

Figure 3 Subdivision of a 15-bit Memory Address

At the centre of the Paging Unit is an Associative Memory called the Page Address Registers. There is one 12-bit Page Address Register associated with each page of core memory. When a memory access is attempted the 21-bit address is sent to the Paging Unit. Here the 12-bit block number is separated from the rest and sent to each of the 64 Page Address Registers in parallel. Each register compares the 12-bit number with its 12-bit contents. If they are different nothing happens. However, if they are the same the Page Address Register sends out its own 6-bit address, this being the number of the page containing this block. The 6-bit page number is combined with the original word number to form a 15-bit memory address (Fig. 4). If the required block is not in core then no Page Address Register will find a match and a Non-Equivalence Interrupt occurs. This results in the One-Level-Store routines being entered to bring the required block into core.

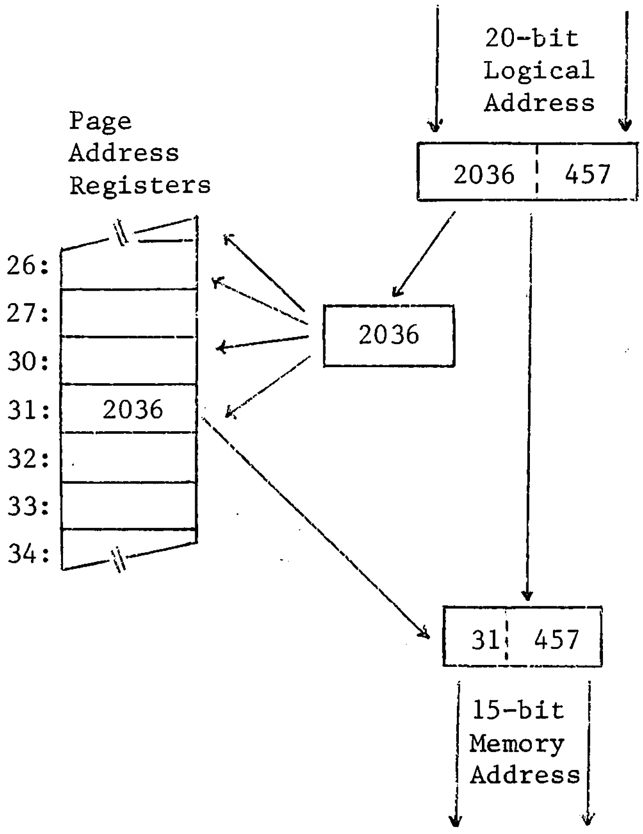

Figure 4 The Atlas Paging Unit